06 | 数据模型无法复用，归根结底还是设计问题

2020-04-15 郭忆

数据中台实战课 进入课程 >

讲述：郭忆
时长 21:59　大小 20.14M

你好，我是郭忆。

上一节课，我带你了解了数据中台如何管理指标，如果我们把指标比喻成一棵树上的果实，那模型就是这棵大树的躯干，想让果实结得好，必须让树干变得粗壮。

先来看一幕真实的场景。

大多数公司的分析师会结合业务做一些数据分析（需要用到大量的数据），通过报表[illegible]服务于业务部门的运营。但是在数据中台构建之前，分析师经常发现自己没有可以复用的数据，不得不使用原始数据进行清洗、加工、计算指标。

由于他们大多是非技术专业出身，写的 SQL 质量比较差，我甚至见过 5 层以上的嵌套。这种 SQL 对资源消耗非常大，会造成队列阻塞，影响其他数仓任务，会引起数据开发的不满。数据开发会要求收回分析师的原始数据读取权限，分析师又会抱怨数仓数据不完善，要啥没啥，一个需求经常要等一周甚至半个月。分析师与数据开发的矛盾从此开始。

这个矛盾的根源在于数据模型无法复用，数据开发是烟囱式的，每次遇到新的需求，都从原始数据重新计算，自然耗时。而要解决这个矛盾，就要搞清楚我们的数据模型应该设计成什么样子。

什么才是一个好的数据模型设计？

来看一组数据，这两个表格是基于元数据中心提供的血缘信息，分别对大数据平台上运行的任务和分析查询（Ad-hoc）进行的统计。

层级	表数量 (已过滤非活跃表)	被读表数量	被写表数量	读表任务数量 (相同任务只算一次)	写表任务数量 (相同任务只算一次)
ODS（原始数据层）	2328	704	1718	1072	923
DWD（明细数据层）	220	140	219	545	220
DWS（轻度汇总层）	63	53	60	187	60
ADS/DM (应用层/集市层)	850	359	825	433	800
DIM（维表）	41	32	27	333	27
合计	3502	1288	2849	1925	2030
未识别层	2547	1778	1461	2095	1185
总表/总任务	6049	3066	4310	3096	3177

极客时间

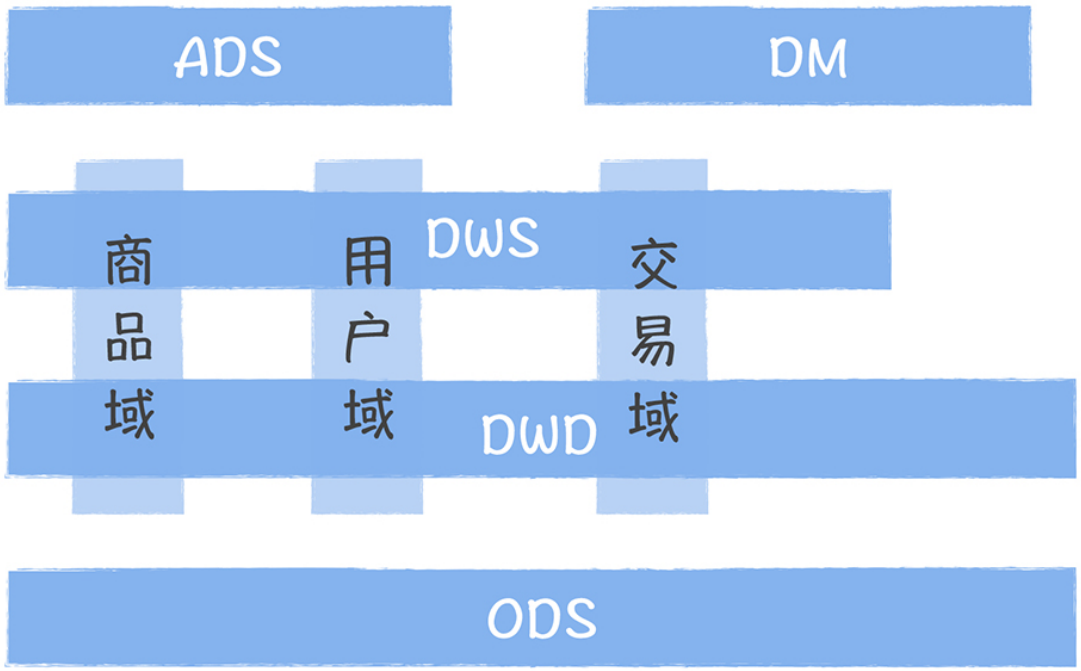
表1 离线调度任务/表统计

层级	Query 命中查询数量
ODS（原始数据层）	892
DWD（明细数据层）	1008
DWS（轻度汇总层）	152
ADS/DM（应用层/集市层）	305
DIM（维表）	862
未识别	909

极客时间

表2 一周内Ad-hoc 查询统计

下图是数仓分层架构图，方便你回忆数据模型分层的设计架构：



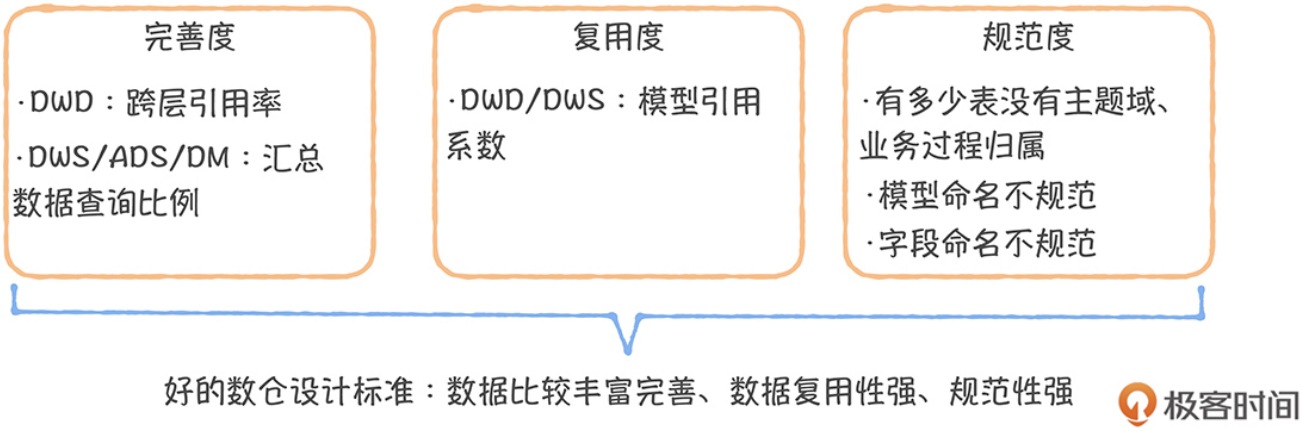
极客时间

我们首先来看表 1。表 1 中有 2547 张未识别分层的表，占总表 6049 的 40%，它们基本没办法复用。 重点是在已识别分层的读表任务中，ODS：DWD：DWS：ADS 的读取任务分别是 1072：545：187：433，直接读取 ODS 层任务占这四层任务总和的 47.9%，这说明有大量任务都是基于原始数据加工，中间模型复用性很差。

我们再来看看表 2，在已识别的分层的查询中，ODS：DWD：DWS：ADS 的命中的查询分别是 892：1008：152：305，有 37.8% 的查询直接命中 ODS 层原始数据，说明 DWD、DWS、ADS 层数据建设缺失严重。尤其是 ADS 和 DWS，查询越底层的表，就会导致查询扫描的数据量会越大，查询时间会越长，查询的资源消耗也越大，使用数据的人满意度会低。

最后，我们进一步对 ODS 层被读取的 704 张表进行分解，发现有 382 张表的下游产出是 DWS，ADS，尤其是 ADS 达到了 323 张表，占 ODS 层表的比例 45.8%，说明有大量 ODS 层表被进行物理深加工。

通过上面的分析，我们似乎已经找到了一个理想的数仓模型设计应该具备的因素，那就是“数据模型可复用，完善且规范”。



如何衡量完善度

DWD 层完善度：衡量 DWD 层是否完善，最好看 ODS 层有多少表被 DWS/ADS/DM 层引用。因为 DWD 以上的层引用的越多，就说明越多的任务是基于原始数据进行深度聚合计算的，明细数据没有积累，无法被复用，数据清洗、格式化、集成存在重复开发。因此，我提出用跨层引用率指标衡量 DWD 的完善度。

> 跨层引用率：ODS 层直接被 DWS/ADS/DM 层引用的表，占所有 ODS 层表（仅统计活跃表）比例。

跨层引用率越低越好，在数据中台模型设计规范中，我们要求不允许出现跨层引用，ODS 层数据只能被 DWD 引用。

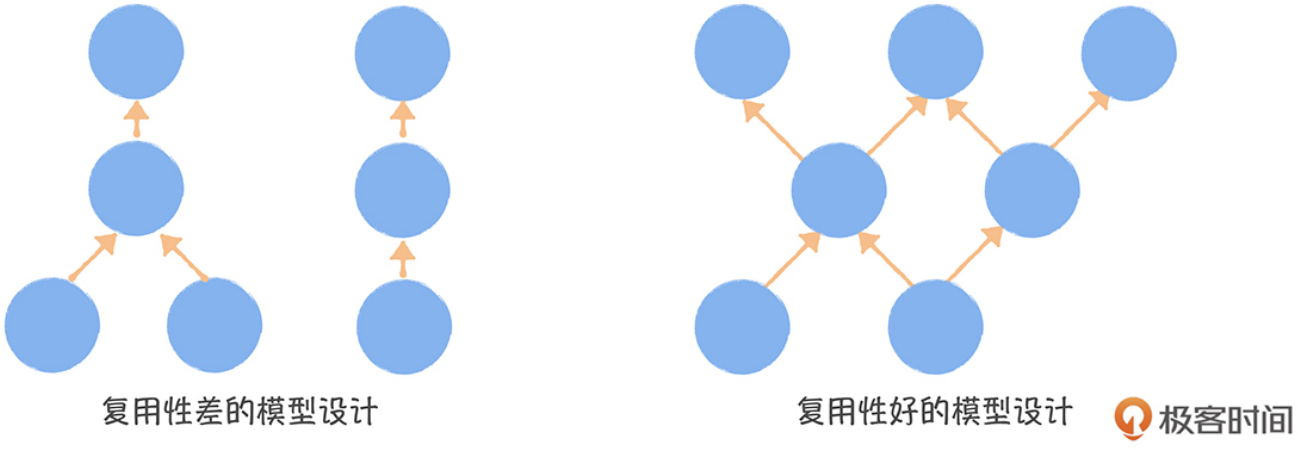
DWS/ADS/DM 层完善度：考核汇总数据的完善度，我认为主要看汇总数据能直接满足多少查询需求（也就是用汇总层数据的查询比例衡量）。如果汇总数据无法满足需求，使用数据的人就必须使用明细数据，甚至是原始数据。

> 汇总数据查询比例：DWS/ADS/DM 层的查询占所有查询的比例。

你要明确的是，这个跟跨层引用率不同，汇总查询比例不可能做到 100%，但值越高，说明上层的数据建设越完善，对于使用数据的人来说，查询速度和成本会减少，用起来会更爽。

如何衡量复用度

数据中台模型设计的核心是追求模型的复用和共享，通过元数据中心的数据血缘图，我们可以看到，一个比较差的模型设计，自下而上是一条线。而一个理想的模型设计，它应该是交织的发散型结构。



我提出用模型引用系数作为指标，衡量数据中台模型设计的复用度。引用系数越高，说明数仓的复用性越好。

> 模型引用系数：一个模型被读取，直接产出下游模型的平均数量。

比如一张 DWD 层表被 5 张 DWS 层表引用，这张 DWD 层表的引用系数就是 5，如果把所有 DWD 层表（有下游表的）引用系数取平均值，则为 DWD 层表平均模型引用系数，一般低于 2 比较差，3 以上相对比较好（经验值）。

如何衡量规范度

表 1 中，超过 40% 的表都没有分层信息，在模型设计层面，这显然是不规范的。除了看这个表有没有分层，还要看它有没有归属到主题域（例如交易域）如果没有归属主题域，就很难找到这张表，也无法复用。

其次，你要看表的命名。拿 stock 这个命名为例，当你看到这个表时，知道它是哪个主题域、业务过程？是全量数据的表，还是每天的增量数据？总的来说，通过这个表名获取的信息太有限了。一个规范的表命名应该包括主题域、分层、表是全量快照，还是增量等信息。

除此之外，如果在表 A 中用户 ID 的命名是 UserID，在表 B 中用户 ID 命名是 ID，就会对使用者造成困扰，这到底是不是一个东西。所以我们要求相同的字段在不同的模型中，它的命名必须是一致的。

讲了这么多，你要如何吸收经验呢？在这里，我给你几点建议。

> 你可以拿着这些指标去评估一下，自己的数仓现状如何。
>
> 然后制订一些针对性的改进计划，比如把这些不规范命名的表消灭掉，把主题域覆盖的表比例提高到 90% 以上。
>
> 在尝试完一段时间的模型重构和优化后，再拿着这些指标去测一测是不是真的变好了。我见过很多数据开发在向上级汇报工作时，喜欢用重构了多少模型说明工作成果，很多老大会想，这些重构到底对数据建设有多少帮助？有没有一些量化的指标可以衡量？**我想有上面的知识，你可以应对这个问题了。**

现在你知道什么是好的数仓设计了，可目前已经存在了大量烟囱式开发，具体怎么做才能让它变成一个数据中台呢？

如何从烟囱式的小数仓到共享的数据中台

建设数据中台本质就是构建企业的公共数据层，把原先分散的、烟囱式的、杂乱的小数仓，合并成一个可共享、可复用的数据中台（我在03 讲提到过，建议你回顾一下）。结合我

在网易的实践经验，我给你几点建议。

第一，接管 ODS 层，控制源头。

ODS 是业务数据进入数据中台的第一站，是所有数据加工的源头，控制住源头，才能从根本上防止一个重复的数据体系的出现。

数据中台团队必须明确职责，全面接管 ODS 层数据，从业务系统的源数据库权限入手，确保数据从业务系统产生后进入数据仓库时，只能在数据中台保持一份。这个可以跟业务系统数据库管理者达成一致，只有中台团队的账号才能同步数据。

ODS 层表的数据必须和数据源的表结构、表记录数一致，高度无损，对于 ODS 层表的命名采用 ODS_ 业务系统数据库名 _ 业务系统数据库表名方式，比如 ods_warehous_stock，warehous 是业务系统数据库名，stock 是该库下面的表名。

第二，划分主题域，构建总线矩阵。

主题域是业务过程的抽象集合。可能这么讲，稍微有点儿抽象，但其实业务过程就是企业经营过程中一个个不可拆分的行为事件，比如仓储管理里面有入库、出库、发货、签收，都是业务过程，抽象出来的主题域就是仓储域。

主题域	描述
商品域	电商平台售卖商品相关的数据，包括商品上架、下架等业务过程
会员域	会员相关的数据，包括注册、登录等业务过程
供应链域	商品采购相关的数据，包括采购、调拨、订单履约等业务过程
物流域	商品配送相关的数据，包括揽收、投递等业务过程
仓储域	商品仓库存储相关的数据，包括入库、出库等业务过程
流量域	电商平台用户相关行为数据，包括搜索、曝光等业务过程
POP域	电商平台的第三方商家数据，包括开店、关店等业务过程
交易域	电商平台交易相关的数据，包括加入购物车、下单、支付等
售后域	电商平台商品售后的相关数据，包括工单、退货等
促销域	商品促销相关的数据，包括使用优惠卷、活动预热等
内容域	电商平台种草、内容社区相关数据，包括发布、分享等业务过程

极客时间

表3 电商业务的主题域划分

主题域划分要尽量涵盖所有业务需求，保持相对稳定性，还具备一定的扩展性（新加入一个主题域，不影响已经划分的主题域的表）。

主题域划分好以后，就要开始构建总线矩阵，明确每个主题域下的业务过程有哪些分析维度，举个例子：

主题域	业务过程	可分析维度					
		商品	用户	类目	省份	流量来源	订单
交易域	加购	Y	Y	Y	Y	Y	Y
	下单	Y	Y	Y	Y	Y	Y
	支付	Y	Y	Y	Y		Y
	确认收货	Y	Y	Y	Y		Y

极客时间

表4 交易域的总线矩阵

第三，构建一致性维度。

售后团队的投诉工单数量有针对地区的分析维度，而配送团队的配送延迟也有针对地区的分析维度，你想分析因为配送延迟导致的投诉增加，但是两个地区的分析维度包含内容不一致，最终会导致一些地区没办法分析。所以我们构建全局一致性的维表，确保维表只存一份。

维度统一的最大的难题在于维度属性（如果维度是商品，那么商品类别、商品品牌、商品尺寸等商品的属性，我们称为维度属性）的整合。是不是所有维度属性都要整合到一个大的维表中，也不见得，我给你几个建议。

公共维度属性与特有维度属性拆成两个维表。在自营平台中，通常也会有一些第三方的商家入驻，但是数量很少。大部分商品其实都没有店铺的属性，这种情况，就不建议将店铺和商品的其他维度属性，比如商品类别、品牌设计成一个维表。

产出时间相差较大的维度属性拆分单独的维表，比如有些维度属性产出时间在凌晨 2 点，有些维度属性产出时间在凌晨 6 点，那 2 点和 6 点的就可以拆成两个维表，确保核心维表尽早产出。

出于维表稳定性产出的考虑，你可以将更新频繁的和变化缓慢的进行拆分，访问频繁的和访问较少的维表进行拆分。

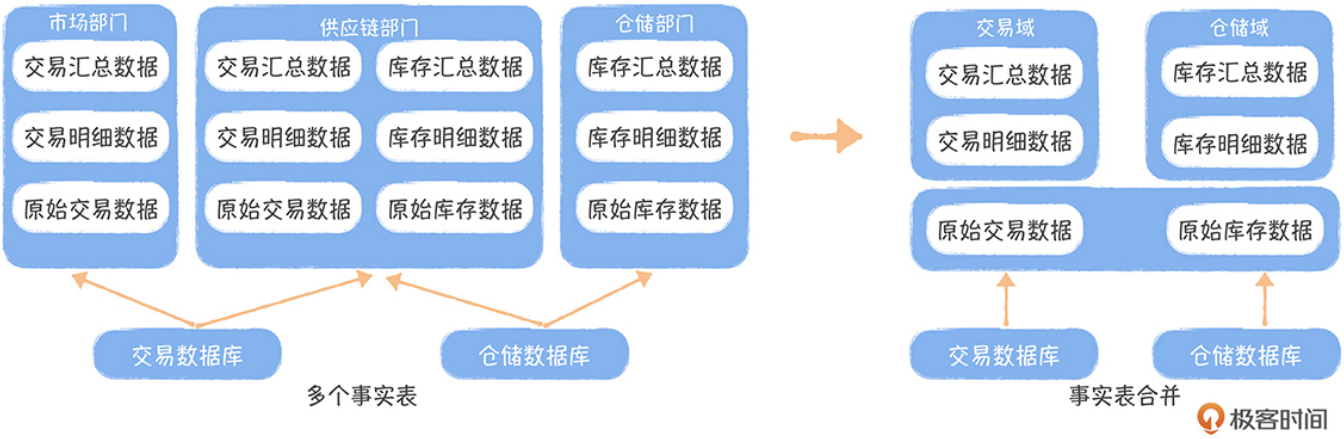
对于维表的规范化命名，建议你用“DIM_ 主题域 _ 描述 _ 分表规则”方式。**分表你可以这样理解：**一个表存储几千亿行记录实在是太大了，所以需要把一个表切割成很多小的分区，每天或者每周，随着任务被调度，会生成一个分区。我提供给你一个常见的分区规则（这个规则你在用的时候，查阅就可以了）。

分表策略	说明
DD	每天分区中保留的是历史至今的全量数据，根据业务使用场景制定例行清理策略
DI	每天分区中保留的是当日的增量数据，可以是汇总数据也可以是明细数据，一般永久保留
WD	每周的分区中保留的是历史至今的全量数据，根据业务使用场景制定例行清理策略
WI	每周的分区中保留的是对应周的增量数据，可以是汇总数据也可以是明细数据，一般永久保留
MI	每月的分区中保留的是对应月份的整个月的增量数据，可以是汇总数据也可以是明细数据，一般永久保留
MD	每月的分区中保留的是历史至今的全量数据，根据业务使用场景制定例行清理策略
ND	死表或者不定期更新的全量表

极客时间

第四，事实表整合。

我觉得事实表整合遵循的最基本的一个原则是，统计粒度必须保持一致，不同统计粒度的数据不能出现在同一个事实表中。来看一个例子：



在数据中台构建前，供应链部门、仓储部门和市场部门都有一些重复的事实表，我们需要将这些重复的内容进行去除，按照交易域和仓储域，主题域的方式进行整合。

字段	类型	约束	备注
sku_id	STRING	唯一	商品ID
storage_id	BIGINT	唯一	仓库ID
stock	BIGINT		库存数量
pre_sale	INT		预售库存

仓储部门库存明细表

统计粒度不同，无法合并

字段	类型	约束	备注
sku_id	STRING	唯一	商品ID
stock	BIGINT		库存数量
alarm_store	INT		预警库存

供应链部门库存明细表

极客时间

对于仓储部门和供应链部门都有的库存明细表，因为仓储部门的统计粒度是商品加仓库，而供应链部门的只有商品，所以原则上两个表是不能合并，而是应该独立存在。

字段	类型	约束	备注
order_id	STRING	唯一	订单ID
buyer_account	STRING		买家账号
trade_no	BIGINT		交易流水号
order_time	BIGINT		下单时间

市场部门下单明细表

字段	类型	约束	备注
order_id	STRING	唯一	订单ID
sku_id	STRING		商品ID
money	BIGINT		交易金额
express_fee	BIGINT		运费

供应链部门下单明细表

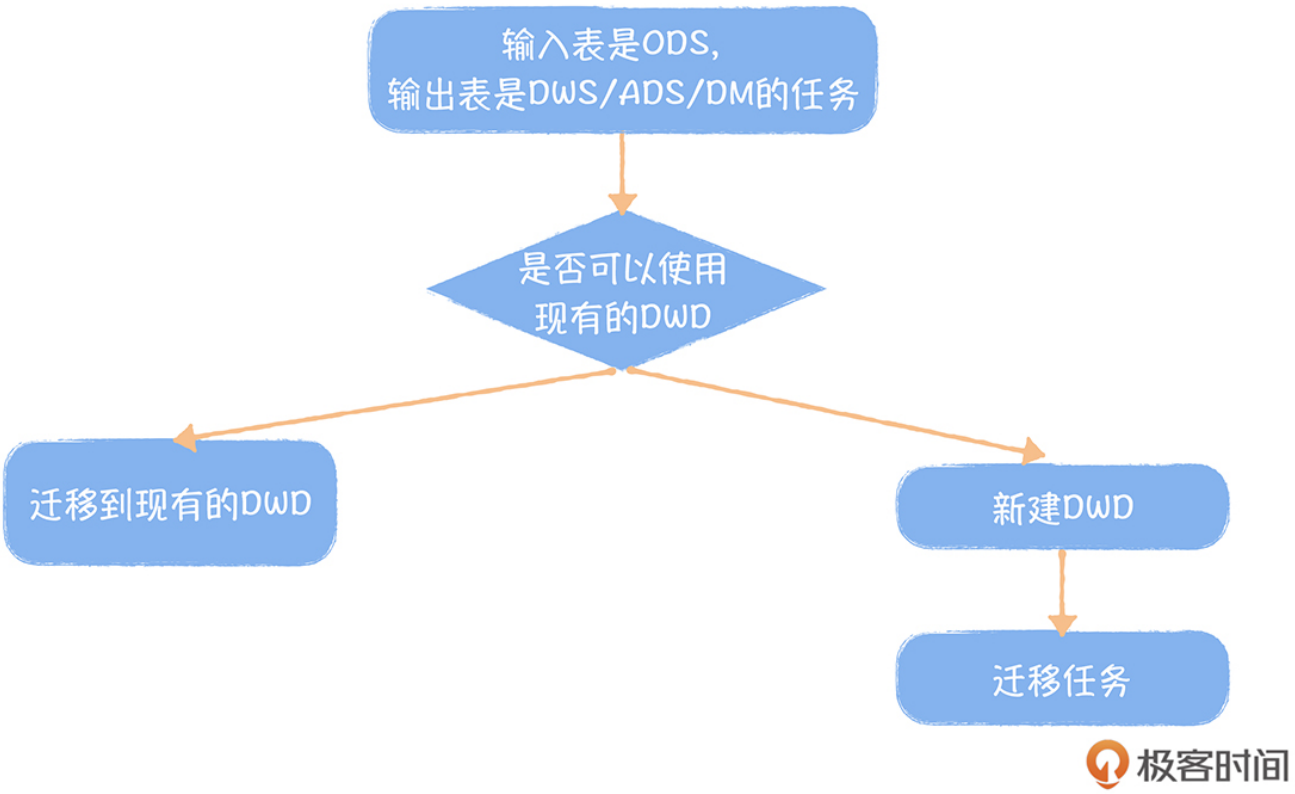
字段	类型	约束	备注
order_id	STRING	唯一	订单ID
buyer_account	STRING		买家账号
trade_no	BIGINT		交易流水号
order_time	BIGINT		下单时间
sku_id	STRING		商品ID
money	BIGINT		交易金额
express_fee	BIGINT		运费

交易域下单明细表

极客时间

对于市场部门和供应链部门的两张下单明细表，因为统计粒度都是订单级别，都归属于交易域下的下单业务过程，所以可以合并为一张事实表。

除此之外，还应该考虑将不全的数据补齐。对于 ODS 层直接被引用产出 DWS/ADS/DM 层的任务，通过血缘，找到任务清单，逐个进行拆解。没有 ODS 对应的 DWD 的，应该生成 DWD 表，对于已经存在的，应该迁移任务，使用 DWD 层表。



极客时间

DWD/DWS/ADS/DM 的命名规则适合采用“[层次][主题][子主题][内容描述][分表规则]”的命名方式。

第五，模型开发。

模型设计完成后，就进入模型开发阶段，我想提几个你需要注意的点：

1. 所有任务都必须严格配置任务依赖，如果没有配置任务依赖，会导致前一个任务没有正常产出数据的情况下，后一个任务被调度起来，基于错误的数据空跑，浪费资源，同时增加了排查故障的复杂度；
2. 任务中创建的临时表，在任务结束前应该删除，如果不删除，会发现有大量的临时表存在，占用空间；
3. 任务名称最好跟表名一致，方便查找和关联；
4. 生命周期的管理，对于 ODS 和 DWD，一般尽可能保留所有历史数据，对于 DWS/ADS/DM 需要设置生命周期，7～30 天不等；
5. DWD 层表宜采用压缩的方式存储，可用 lzo 压缩。

第六，应用迁移。

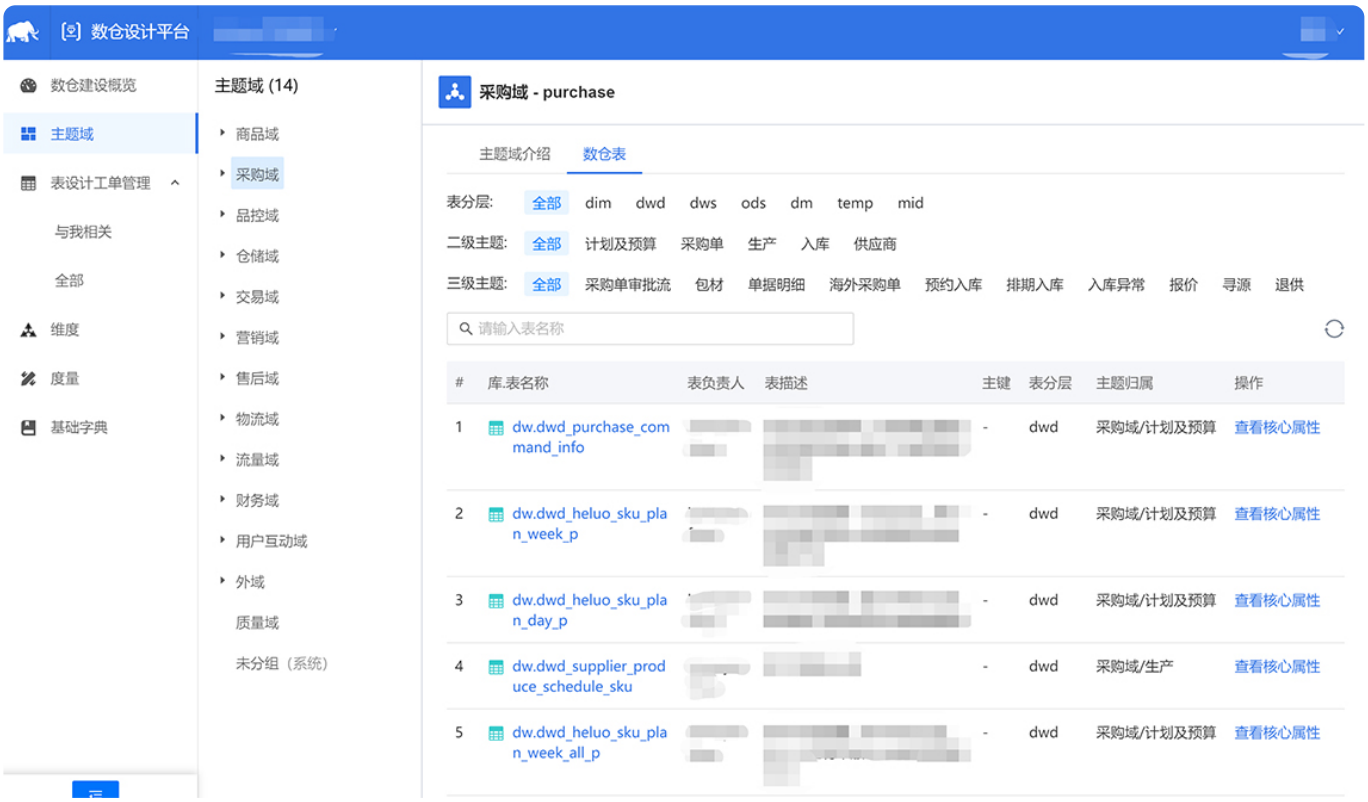
最后一步就是应用的迁移，这个过程的核心是要注意数据的比对，确保数据的完全一致，然后进行应用迁移，删除老的数据表。

总的来说，建设数据中台不是一口气就能吃成一个胖子，它的建设往往是滚雪球的方式，随着一个个应用的迁移，中台的数据也越来越丰满，发挥的价值也越来越大。

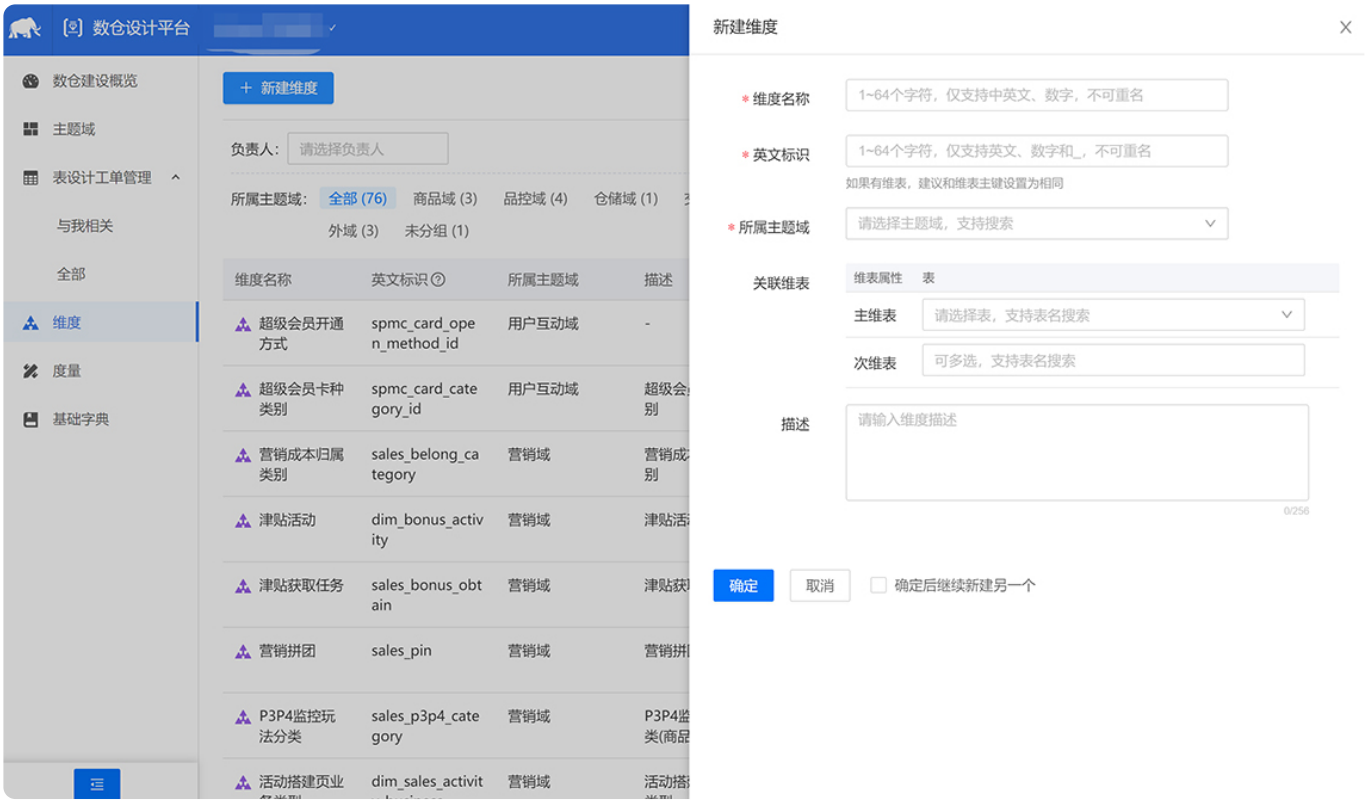
数仓建模工具 EasyDesign

当然了，这些步骤的实现，离不开一个好用的工具作为支撑，为了规范化数据模型的设计，我们研发了 EasyDesign 的模型设计产品，让这些流程实现系统化管理。而我希望你了解如何去设计这个工具，或者在选用工具的时候应该考虑有哪些功能。

EasyDesign 构建于元数据中心之上，通过 API 调用元数据中心的数据血缘接口，结合数仓模型设计的指标，给出了模型设计度量。



EasyDesign 按照主题域、业务过程、分层的方式管理所有的模型。



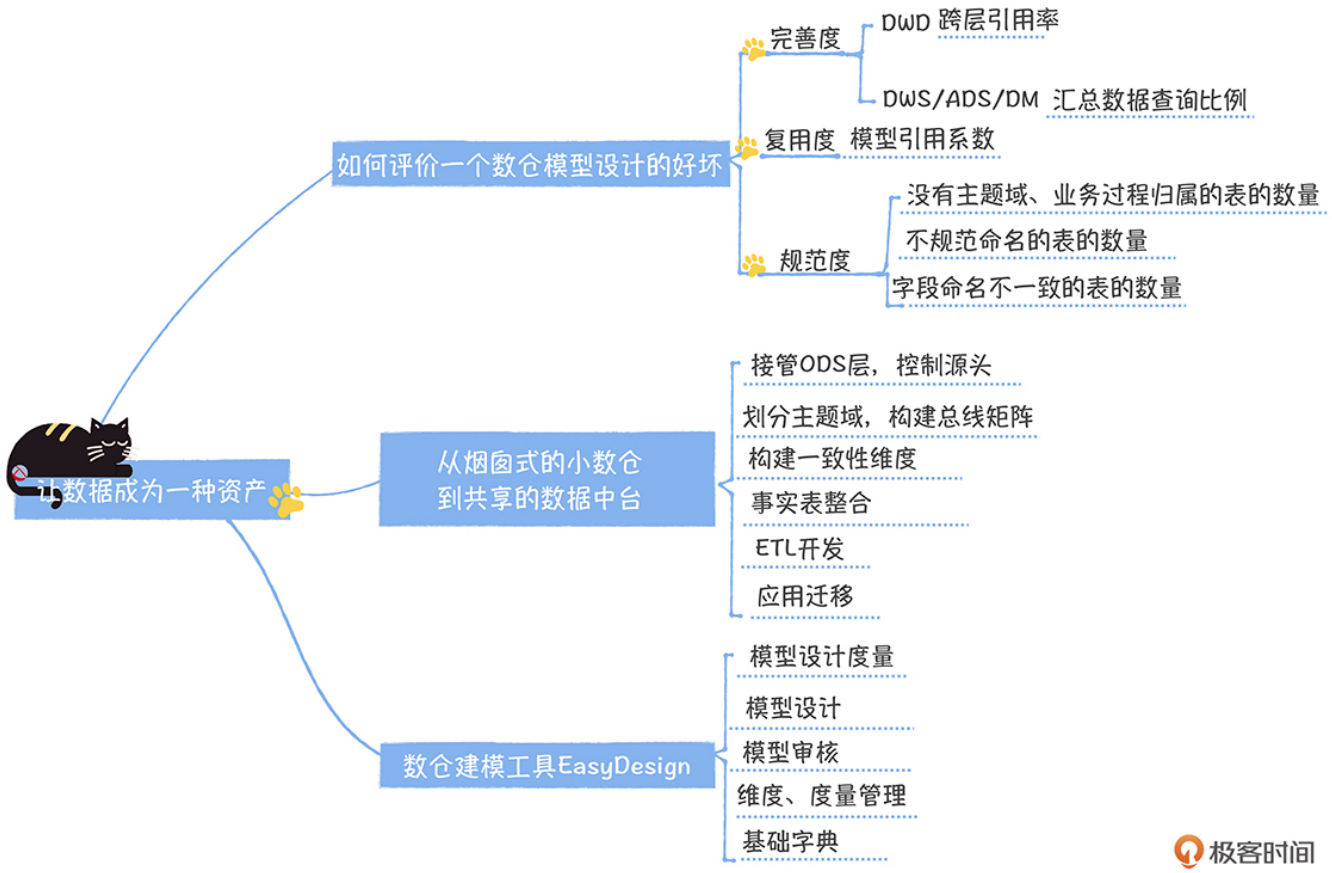
它还提供了维度、度量和字段基础字典的管理，同时具备模型设计审批流程的控制。

课堂总结

本节课，我结合自己的经验，带你了解了数据中台的模型设计。从确立设计目标，到通过一系列步骤，将一个个分散的、杂乱的、烟囱式的小数仓逐步规整到一个可复用、可共享的数据中台，最后通过产品化的方式实现系统化的管理。在最后，我想再强调几个点：

> 完善度、复用度和规范度构成了衡量数据中台模型设计的度量体系，可以帮助你评估数仓设计的好坏。
>
> 维度设计是维度建模的灵魂，也是数据中台模型设计的基础，维度设计的核心是构建一致性维度。
>
> 事实表的统计粒度必须保持一致，不同统计粒度的数据不能出现在同一个事实表中。

你要知道，数据中台的构建往往需要花费半年甚至一年以上的时间，但是数据中台建成后，对研发效率的提升效果非常明显，在网易电商业务中，中台构建后相比构建前，数据需求的平均交付时间从一周缩短到 3 天内，需求响应速度的提升，为企业运营效果提升提供了数据支撑。我相信通过数据中台的构建，你所在企业的数据研发效率也会得到大幅度的提升。



思考时间

在数据中台实际实施落地的过程中，数据团队不但要建设公共数据层，形成数据中台，还要承担着巨大的新需求的压力。而且，往往需求的优先级都高于建设公共数据层的优先级，导致中台建设的进度难以保障。 对这个问题，你有什么解决方法呢？

最好，感谢你的阅读，如果这节课让你有所收获，也欢迎你将它分享给更多的朋友。

© 版权归极客邦科技所有，未经许可不得传播售卖。 页面已增加防盗追踪，如有侵权极客邦将依法追究其法律责任。

上一篇　05 | 如何统一管理纷繁杂乱的数据指标?

下一篇　07 | 同事老打脸说数据有问题，该怎么彻底解决?

精选留言 (17)

写留言

大尾巴老猫
2020-04-15

1、先满足需求（活下去），再研发公共数据层（构建美好未来）。
2、获得高层领导的支持，以获得更多的研发资源。
3、在满足业务需求的过程中，根据业务需求不断对公共数据层进行迭代和优化。
4、随着时间的推移，越来越多的日常业务需求可以用公共数据层（中台来完成）。
5、日常业务需求开发和公共数据层构建是相互促进的循环。

展开

作者回复: 不错，总结的挺好的，其实我只想再补充一点，就是为了保障数据中台的推进速度，可以尝试成立专人团队，这些人的目标明确就是中台构建，模型的重构和整合，指标的梳理。这些人不接业务需求，这样可以避免日常业务需求对数据团队的中台建设的干扰。否则的话，数据中台的建设进度，经常会受到业务需求压力的干扰，而且如果没有明确的KPI，或者KPI权重不够大，中台建设的动力也会不足。

感谢你的阅读，总结的很棒！

3

麻婆豆腐
2020-04-15

郭老师好，请教一个小白的问题，运行任务和分析查询的统计是怎么统计的呢？我们现在的平台有用hive直接查的，有tableau连impala查询的，还有spark定时任务，各种各样的，怎么才能统计出像您这样的表格呢。也想做下自己的模型指标分析。十分感谢！

展开

作者回复: 你好，可以基于数据血缘来实现，一个表的产出任务以及它的下游引用任务，数据血缘都是有的。

对于分析查询，目前我们有两个平台，一个是网易有数，类似tableau，一个是自助分析平台，就是执行SQL的，我们把这两个平台的日志执行信息会拿出来进行离线的分析和统计，然后去看每个query查询了哪些表。

如果你是tableau，可能没这么方便，不过可以试试从impala入手，impala侧日志中是有SQL信息，可以抓出来分析统计。对于spark和hive，可以基于数据血缘来实现。

感谢你的阅读~

1 2

泡泡鱼大王
2020-04-16

首先高优先级需求一定是先开发。
我觉得压缩空间在中台项目本身，
1.尽早搭建共享数仓部分。
措施：
技术上把各个小数仓的元信息和数据模型 通过自动化采集到同一个数据库中，进行分析...

展开

作者回复: 说的好，中台对数据开发的要求真的是挺高的，当然，也需要一些好用的工具产品来降低他们的工作量和复杂度，一个优秀的PM当然也是必须的。

我一直强调一个观点，不要用技术的思维看待数据中台，要用管理的思维，数据中台它是一个系统性的工程，对整个数据建设是一次革命！

感谢你的阅读，期待与你在留言区再次相遇，也期待你继续发表你的观点，很棒！

1

Samuel

2020-04-16

郭老师,刚接触数据中台,提的问题可能还比较初级,维表是一个基于的事实表的维表吗?他们之间是什么关系?需要ID关联吗?还是放到一张大的宽表中?多谢了!

展开

作者回复: 你好，我来回到一下你的问题。

维表和事实表，可以通过ID关联，构建星型模型，甚至维度本身也可以构建有层次关系的星座模型。

在数据中台的集市层，或者汇总层，为了方便使用者使用，我们会做一些维度退化的设计，降低Join的次数，因为通过id关联，必须就会是涉及到join，join操作对计算引擎的性能要求比较高，所以会把一些常用的维度属性退化到事实表中，这个还是主要取决于业务场景和需求。

感谢你的阅读，期待与你再次相遇~

1

Geek_9319

2020-04-15

一致性维度能否详细解释下，有点不太理解

展开

作者回复: 你好，我来回到一下你的问题。

在业务系统中，可能会存在用户表不一致的情况，那如果到数据中台来，我们就必须要构建统一的用户维表，不能存在多个用户维表，那这就导致我们没办法做一致性上卷，基于公共维度，实现多个事实的关联查询分析了。

所以在数据中台中，我们必须要统一维度，构建一致性维表。

感谢你的阅读，期待与你再次相遇~

北野豪横
2020-04-15

目前在做一个全新的领域，警务中台系统，跟原本的电商模式有着很大差别，本来一头雾水的项目，读了老师的课有点云开见月明的意思，提出一点个人的感想，不论是模型，还是输出的指标，个人感觉越来越多的应该从底层业务出发，自底向上来驱动整个业务中台，这里需要模型与业务与数据的多项循环反馈机制才能逐渐完善整个中台，如何将指标模块化，如何让各种模型即产中中间层结果，又产出直接结果，形成真正的积木式中台...

展开

作者回复: 首先，感谢你的认可。看到对你的实际工作有所帮助和启发，我感到很高兴。

你说的非常正确，业务和数据中台之间本来就要相互反馈，业务人员经常去查一些明细表，那我们中台就要考虑，是不是需要完善汇总层的模型。

你说数据中台对从业人员要求高，我也非常赞同，数据中台从业务中来，最终必须要回到业务。数据中台团队独立于业务，但是绝对不能脱离业务。我在第13讲中，会详细的介绍，感谢你在留言区发表的想法，非常棒！

期待与你再次相遇~

1

JohnT3e
2020-04-15

目前公司数据中台建设已有一年有余，今天的内容有很多共鸣之处，收获颇多。

作者回复: 谢谢你的鼓励，期望能够对你后续的工作有所帮助和启发~

1

3.3
2020-04-23

请教，在主题域这一层，老师认为是范式建模好，还是范式模式？现在很多银行传统数仓的主题层都是三范式建模。

展开

小崔

2020-04-23

对于维表一直有疑惑。请问维表是否就是一张大宽表，作用是根据id和具体记录关联起来，在生成DWD表记录时补全记录缺失的维度值？麻烦老师解答。

展开

毛聪

2020-04-22

郭老师好，我想问个模型分层的问题，我计算的一个指标从dwd层到dws层后就可以直接给到应用层了，那这个汇总表到底算是dws层的表还是ads层的表？

展开

小桥流水

2020-04-19

领导安排落地数仓分层定义及可实在落地 可度量 在这找到答案了

1、明确每层都有哪些表

2、通过分析执行sql知道 哪些表的使用情况 计算出 使用率

3、指标体系建设明确了业务指标和表的关系

4，血缘关系分析 明确了表之间的关系...

展开

作者回复: 非常高兴可以帮助到你~

感谢你的阅读，期待与你再次相遇~

rageeeee

2020-04-17

老师，维度表的分表策略这个没有看懂，我理解维度表的数据应该是变化不大的，为什么还分那么多区，您能详细的再解释一下吗？

展开

作者回复: 不一定哦，维表不一定是变化不大的，例如用户维表，相比事实表，确实变化算小的，但是并不是说维表是不变化的。每天还有有很多新增注册用户的！

感谢你的阅读，期待与你再次相遇~

从前有座山
2020-04-17

郭老师，您好，非常感谢这么精彩的分享。
有个问题请教，关于“一致性维度”的，业务场景如下：企业X有A和B两个系统，都有客户表（数据有部分交集（根据一定规则例如名称相同判断的），没有统一的客户中心），数据中台中客户维度是否只有一个？客户表的维度歧义处理是否是数据中台的职责？

展开

作者回复: 你好， 首先要感谢你的认可和鼓励。

我来谈谈你的问题。在数据中台中，用户维表必须要统一，不同业务系统用户不统一，在构建数据中台统一用户维表时，需要由中台部门来解决，可以使用ID-Mapping技术，打通不同系统之间的账号，建立统一的用户维表。

感谢你的阅读，期待与你再次相遇~

幸福时光
2020-04-16

非常具有实战的指导意义，感谢老师分享

展开

作者回复: 感谢你鼓励和认可，希望对你有所帮助，这些经验也确实是我们在过去一年中沉淀和总结下来的，比较有实践指导意义。

如果后续你有什么问题，可以直接在留言区与我交流。再次感谢你的阅读，下次见~

aof
2020-04-15

老师文章里面说的每一点都有比较明确的实施意义！赞！

作者回复: 感谢你的阅读，也谢谢你的鼓励。

吴科

2020-04-15

任何情况下，都是以业务需求为主。数仓模型的优化，在完成需求的情况下，一步步优化。
老师，说的回收ods层数据，太对了，避免重复计算，模型重复建设。
维度建模方面，老师能详细讲解个案例吗，我们现在建模还不规范，想参考一下。

展开

作者回复: 你好，管住ODS，就掌握了主动权。

为了保证数据中台的推进速度，我反而建议成立独立的中台重构小团队，否则很容易受到业务需求的挤压。

至于维度建模的方法，后续我可以拿个案例与你来具体分享。

感谢你的阅读~

leslie

2020-04-15

这种解决方案其实是项目管理的事情，雷蓓蓓老师的《项目管理实战20讲》学完之后觉得所谓的安排已经一切从容多了；如何去合理安排每个项目的时间和工作完成度。。。相关的策略是关键。
换个方式去讲吧。我手上既有数据系统架构和代码优化的工作，同时还有运维系统优化的工作；如何合理去安排其实是我每天早上或者每个周期会去规划的事情，合理规划好自...

展开

作者回复: 蓓蓓，我12年做网易分布式数据库项目的时候，就是我们项目的PM啦。数据中台是一项系统性的工作，自然项目管理也是不可或缺的一部分。

感谢你的阅读~